S&H Form: FORM PTO-1390 (2/01)

JC07 Rec'd PCT/PTO 19 FEB 2002

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY'S DOCKET NUMBER 1454.1223
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371	

INTERNATIONAL APPLICATION NO. PCT/DE00/02705	INTERNATIONAL FILING DATE 11 August 2000	PRIORITY DATE CLAIMED 18 August 1999

TITLE OF INVENTION
AGENT SYSTEM FOR MOBILE AGENTS, COMPUTER NETWORK AND METHOD FOR DOWNLOADING AN AGENT SYSTEM FROM A HOST COMPUTER TO A CLIENT COMPUTER OF A COMPUTER NETWORK

APPLICANT(S) FOR DO/EO/US
Birgit KRELLER et al.

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.
2. ☒ This is an express request to immediately begin national examination procedures (35 U.S.C. 371(f)).
3. ☒ The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
4. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
    a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
    b. ☐ has been transmitted by the International Bureau.
    c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
5. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
6. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
    a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
    b. ☐ have been transmitted by the International Bureau.
    c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US)
7. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
8. ☒ An oath or declaration of the inventor (35 U.S.C. 371(c)(4)).
9. ☐ A translation of the Annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 10-15 below concern document(s) or information included:

10. ☒ An Information Disclosure Statement Under 37 CFR 1.97 and 1.98.
11. ☒ An assignment document for recording.
    Please mail the recorded assignment document to:
    a. ☒ the person whose signature, name & address appears at the bottom of this document.
    b. ☐ the following:
12. ☒ A preliminary amendment.
13. ☒ A substitute specification
14. ☐ A change of power of attorney and/or address letter.
15. ☒ Other items or information: First page of published International Application with translation.

☒ The U.S. National Fee (35 U.S.C. 371(c)(1)) and other fees as follows:					
CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS	25 -20=	5	x $ 18.00	90.00
	INDEPENDENT CLAIMS	4 -3=	1	x $ 84.00	84.00
	MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+$280.00	0.00

BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(4):
☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO $1,040
☒ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO.........$ 890
☐ International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but international search fee (37 C.F.R. 1.445(a)(2) paid to USPTO...$ 740
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provision of PCT Article 33(1)-(4)..........$ 710
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2) to (4) ....$ 100

890.00

Surcharge of $130 for furnishing the National fee or oath or declaration later than ☐ 20 ☐ 30 mos. from the earliest claimed priority date (37 CFR 1.482(e)).

0.00

TOTAL OF ABOVE CALCULATIONS	1,064.00

Reduction by 1/2 for filing by small entity, if applicable. Affidavit must be filed also. (Note 37 CFR 1.9, 1.27, 1.28.)

SUBTOTAL	1,064.00

Processing fee of $130 for furnishing the English Translation later than [ ] 20 [ ] 30 mos. from the earliest claimed priority date (37 CFR 1.482(f)).

TOTAL NATIONAL FEE	1,064.00
Fee for recording the enclosed assignment (37 CFR 1.21(h)).	+40.00
TOTAL FEES ENCLOSED	1,104.00

a. ☐ A check in the amount of $ to cover the above fees is enclosed.
b. ☒ Please charge my Deposit Account No. 19-3935 in the Amount of $1,104.00 to cover the above fees. A duplicate copy of this sheet is enclosed.
c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required,
or   credit any overpayment to Deposit Account No. 19-3935. A duplicate copy of this sheet is enclosed.

21171
PATENT TRADEMARK OFFICE

SUBMITTED BY: STAAS & HALSEY LLP

Type Name	Mark J. Henry	Reg. No.	36,162
Signature	*[signature]*	Date	Fb 19 2002

Docket No.: 1454.1223

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Birgit KRELLER et al.

Serial No.: (PCT/DE00/02705)   Group Art Unit: To be assigned

Filed: Examiner: To be assigned

For: AGENT SYSTEM FOR MOBILE AGENTS, COMPUTER NETWORK AND METHOD
        FOR DOWNLOADING AN AGENT SYSTEM FROM A HOST COMPUTER TO A
        CLIENT COMPUTER OF A COMPUTER NETWORK

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Before examination of the above-identified application, please amend the application as

follows:

IN THE ABSTRACT:

Please REPLACE the Abstract originally filed with the enclosed Substitute Abstract

attached hereto.

IN THE SPECIFICATION:

Please REPLACE the specification originally filed with the enclosed Substitute

Specification.

IN THE CLAIMS:

Please CANCEL claims 1-15.

Please ADD new claims 16-40 in accordance with the following:

16. (NEW) An agent system which can be installed on computers connected in a

computer network, comprising:

an agent platform which comprises service programs required in order for the agent

system to be executed on the respective computers,

an agent system controller which comprises an agent system launcher to be loaded

onto a client computer independently of a reminder of the agent system and is used for loading the agent system from a host computer, such that, before the agent system is loaded from the host computer onto the client computer, the agent system launcher sends an identifier identifying at least one of client computer hardware and client computer software, the identifier being sent to the host computer, and

an agent system update program to be installed on the host computer to adjust the agent system based on the identifier and at least one of client computer hardware and client computer software, and then to transmit the agent system to the client computer .

17. (NEW) The agent system as claimed in claim 16,

wherein the agent system launcher is designed both for loading a full agent system and for loading parts of the agent system .

18. (NEW) The agent system as claimed in claim 16,

wherein the agent system launcher tests at least one of hardware and software of the client computer in order to produce the identifier.

19. (NEW) The agent system as claimed in claim 16,

wherein the agent system launcher compares a version of a previous agent system installed on the client computer with a version of the agent system which can be loaded from the host computer .

20. (NEW) The agent system as claimed in claim 16, wherein the agent system is an agent system for mobile agents.

21. (NEW) The agent system as claimed in claim 17,

wherein the agent system launcher tests at least one of hardware and software of the client computer in order to produce the identifier.

22. (NEW) The agent system as claimed in claim 21,

wherein the agent system launcher compares a version of a previous agent system installed on the client computer with a version of the agent system which can be loaded from the host computer.

2

23. (NEW) The agent system as claimed in claim 22, wherein the agent system is an agent system for mobile agents.

24. (NEW) A computer network having a plurality of computers, comprising:

a host computer;

a client computer;

data lines connecting the host and client computers; and

an agent system comprising:

an agent system launcher loaded onto the client computer to send to the host computer an identifier identifying at least one of client computer hardware and client computer software and to subsequently load the agent system onto the client computer, and

an agent system update program installed on the host computer to adjust the agent system based on the identifier and at least one of client computer hardware and client computer software, and then to transmit the agent system to the client computer.

25. (NEW) The computer network as claimed in claim 24,

wherein at least one of the computers is a mobile computer.

26. (NEW) The computer network as claimed in claim 25,

wherein the mobile computer is connected to the computer network by a nonpermanent data line .

27. (NEW) The agent system as claimed in claim 24,

wherein the agent system launcher is designed both for loading a full agent system and for loading parts of the agent system.

28. (NEW) The agent system as claimed in claim 24,

wherein the agent system launcher tests at least one of hardware and software of the client computer in order to produce the identifier.

29. (NEW) The agent system as claimed in claim 24,

wherein the agent system launcher compares a version of a previous agent system installed on the client computer with a version of the agent system which can be loaded from the host computer.

3

30. (NEW) The agent system as claimed in claim 24, wherein the agent system is an agent system for mobile agents.

31. (NEW) A method for loading a system from a host computer to a client computer, comprising:

agent system launcher sending an identifier identifying at least one of hardware and software of the client computer, the identifier being sent from an agent system launcher loaded on the client computer, the identifier being sent to the host computer ;

compiling an agent system adjusted to at least one of hardware and software of the client computer based on the identifier received at the host computer from the client computer, the agent system being compiled by an agent system update program installed on the host computer;

transmitting the agent system to the client computer ; and

loading the agent system on the client computer by the agent system launcher.

32. (NEW) The method as claimed in claim 31, wherein

the agent system launcher first checks whether a previous agent system is already installed on the client computer,

if the previous agent system is installed, the agent system launcher tests whether an update is possible or required and whether the previous agent system is in operation, and

if an update is possible or required and the previous agent system is in operation, the operation of the agent system is terminated so that it can be updated.

33. (NEW) The method as claimed in claim 31, wherein a call to the host computer involves the client computer sending the identifier to the host computer and then receiving the agent system from the host computer .

34. (NEW) The method as claimed in claim 32, wherein a call to the host computer involves the client computer sending the identifier to the host computer and then receiving the agent system from the host computer.

35. (NEW) The method as claimed in claim 33, further comprising:

if a previous agent system is loaded on the client computer, determining whether the

4

previous agent system should be updated; and

updating the previous agent system .

36. (NEW) The method as claimed in claim 35, further comprising:

testing whether an update is possible in principle;

if an update is possible, sending a version identifier to the host computer ;

sending to the client computer version information indicating a most recent agent system which can be loaded from the host computer ;

comparing the version identifier with the version information, and

if the comparison reveals that the version which can be loaded from the host computer is more up to date than the version of the previous agent system which is on the client computer, updating the previous agent system.

37. (NEW) The method as claimed in claim 36,

wherein comparing the version identifier with the version information involves testing whether separable parts of the agent system on the host computer are more up to date than corresponding separable parts of the previous agent system on the client computer, and

if separable parts on the host computer are more up to date, then only the separable parts of the previous agent system are updated .

38. (NEW) The method as claimed in claim 31,

wherein the identifier comprises information on the client computer selected from the group consisting of memory size, type of display, typical processing speed, type of network and system software used.

39. (NEW) The method as claimed in claim 37,

wherein the identifier comprises information on the client computer selected from the group consisting of memory size, type of display, typical processing speed, type of network and system software used.

40. (NEW) A method for loading a system from a host computer to a client computer, comprising:

sending an identifier from the client computer to the host computer, the identifier identifying at least one of client computer hardware and client computer software;

5

adjusting an agent system to at least one of client computer hardware and client computer software based on the identifier, the agent system being adjusted at the host computer;

transmitting the agent system to the client computer; and

loading the agent system on the client computer.

REMARKS

This Preliminary Amendment is submitted to improve the form of the specification as originally-filed. A substitute specification and marked-up copy of the original specification are enclosed. No new matter is added to these documents.

It is respectfully requested that this Preliminary Amendment be entered in the above-referenced application.

If any further fees are required in connection with the filing of this Preliminary Amendment, please charge same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: _Feb. 19, 2002_   By: _Mark Henry_

Mark J. Henry
Registration No. 36,162

700 Eleventh Street, NW, Suite 500
Washington, D.C. 20001
(202) 434-1500

6

SUBSTITUTE ABSTRACT

An agent system which can be loaded onto a computer in a network is adjusted based on the hardware and/or software of the computer. This makes it possible to incorporate even small mobile computers, in particular simple process computers, into a network in which agent systems are installed on the individual computers, without the power of the agent system needing to be significantly restricted in principle and without needing to dispense with the fundamental applicability of agent systems for small computers.

7

SUBSTITUTE SPECIFICATION

TITLE OF THE INVENTION

AGENT SYSTEM FOR MOBILE AGENTS, COMPUTER, NETWORK AND METHOD FOR DOWNLOADING AN AGENT SYSTEM FROM A HOST COMPUTER TO A CLIENT COMPUTER OF A COMPUTER NETWORK

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is based on and hereby claims priority to PCT Application No. PCT/DE00/02705 filed on August 11, 2000, German Patent Application No. 199 39 058.4 filed on August 18, 1999, the contents of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] The invention relates to an agent system, in particular an agent system for mobile agents, to a computer network having a plurality of computers networked by data lines and to a method for loading an agent system from a host computer onto a client computer.

[0003] Agents are autonomous, cooperative software units comprising code and data. They are independently operating software units which require no continual interaction with the user. There are both static and mobile agents.

[0004] Mobile agents are known, by way of example, from US 5,603,031 and from the book Software Agents, Bradshaw Jeffrey, Menlo Park, California: AAAI Press/The MIT Press, 1996. The chapter on mobile agents in this book can be retrieved in a slightly modified version on the Internet at the address http://www.generalmagic.com/technology/techwhitepaper.html.

[0005] Mobile agents are programs which can be active at various stations in a computer network and can change their location in the computer network. Conventionally, programs in a network are loaded from one computer onto another computer while they are not active. By contrast, mobile agents are first activated and automatically change their position in the network while they are active.

[0006] This small but fundamental difference with respect to conventional programs has considerable effects on the type of applications on the mobile agents in the network. The agents are created at a particular station in the network, which involves stipulating the functions needing to be performed by the agent, so that the agent, having been transmitted to another station in the network, represents the previous station in the network and performs the

appropriate functions. This means that functions can be stipulated at one station in the network which are performed at another station in the network. In principle, this is also possible with conventional systems, with this requiring a permanent exchange of information in the network between the two stations so that the station at which the appropriate control operations are stipulated transmits them to the station at which they are performed.

[0007] By contrast, the mobile agents require no such control operations, since their functions are stipulated once in advance and are then performed automatically. This significantly reduces the data throughput in the network, with any desired operations being able to be performed at the stations remote from the task issuer.

[0008] However, the agents do not always need to operate fully detached from their initiator, but rather can set up communication channels thereto or also to other stations in the network.

[0009] When moving from one station in a network to another station in the network, the mobile agents execute a migration command (e.g. "go"). Such a command requires a destination description containing the name, address or type of the station to which the agent is intended to move, or a combination of these. Different stations do not always need to be arranged at different computers in the network. It is also possible for agents to be active at different stations within one computer.

[0010] Mobile agents can also multiply, so that they perform their functions independently of one another at different stations.

[0011] Mobile agents are frequently used for collecting predetermined data, some of which are evaluated in situ, so that an already preprocessed result is sent to the original task issuer.

[0012] One possible application for such mobile agents arises, by way of example, when a purchaser of a particular object receives a low price guarantee from the vendor. The purchaser can then send out one or more mobile agents which independently obtain corresponding offers from other suppliers, possibly monitor them over a certain period of time and send them to the purchaser, so that he can compare them with the price offered by the vendor.

[0013] Other, known mobile agents and agent systems are marketed under the names Aglets, Voyager and Concordia and can generally be obtained over the Internet.

[0014] So that the mobile agents on the individual computers in the network can be active,

these computers need to have a respective "agent system". An agent system is a collection of service programs which need the respective agents for their tasks. These service programs generally comprise an agent manager connected to other service programs, so that the active mobile agents can fall back on the respective computer resources and can migrate from one agent system in the network to another agent system in the network via network interfaces. Generally, an agent system controller is provided which regulates the execution of the agent system on the respective computer.

[0015] These agent systems form the stations in the network at which the agents can become active.

[0016] Such agent systems are generally written in a platform-independent computer language, such as JAVA, so that they can run on any computer systems.

[0017] The transfer of the agents in the network while they are active creates a new programming paradigm according to which the program code itself is moved to the data needing to be processed instead of data being transported to the processing code or the processing code needing to be loaded in the inactive state onto the respective computer.

[0018] Agents are generally relatively small programs which are sent out and exploit the intelligence available in situ. Since mobile agents can also perform their tasks when there is no permanent network connection between the individual computers, they are ideally suited to use on mobile computers, which are generally not permanently connected to a network.

[0019] However, the agent systems known to date are too extensive for them to be able to be installed on arbitrary simple mobile computers.

[0020] Within the context of the present patent application, the term computer also covers simple process computers equipped merely with a minimal hardware configuration (processor, memory and display). Such simple process computers are integrated on a single chip, for example.

SUMMARY OF THE INVENTION

[0021] One possible object of the invention is to provide an agent system which is able to offer the scope of functions of known agent systems and is nevertheless also suitable for small computers, such as small process computers.

[0022] Another possible object of the invention is to provide an appropriate computer network and a method for loading an agent system from a host computer onto a client computer in a computer network.

[0023] The agent system, which can be installed on computers connected in a computer network, has an agent platform and an agent system controller. The agent platform comprises service programs required by a mobile agent in order to be able to be executed on the respective computer. The agent system controller contains an agent system launcher which can be loaded onto a client computer independently of the rest of the agent system and is used for loading the agent system from a host computer.

[0024] The agent system is distinguished in that the agent system launcher is designed such that, before an agent system is loaded from a host computer onto the client computer on which at least the agent system launcher is installed, an identifier describing the hardware and/or software of the client computer is sent to the host computer, and a server system, the agent system update program, installed on the host computer is designed such that it takes the identifier describing the hardware and/or the software of the client computer as a basis for compiling an agent system adjusted to the hardware and/or software of the client computer and loads this agent system onto the client computer.

[0025] Within the context of the present invention, a host computer is any computer in the network from which it is possible to load an agent system onto another computer, which is referred to as a client computer . It is thus possible for any computer in the network to be a host computer and a client computer, irrespective of its function otherwise in the network.

[0026] The agent system launcher, which sends an identifier identifying the client computer to the host computer, and the agent system which is installed on the host computer and takes this identifier as a basis for compiling an adjusted agent system and makes this adjusted agent system available to the client computer make it possible for even small, in particular mobile, computers, for example comprising a simple process computer or a personal digital assistant, to be incorporated into a computer network and to be provided with the agent system, since this agent system is automatically adjusted to the respective client computer and its power.

[0027] Typical criteria for adjustment are, by way of example, the display (e.g. color picture tube, color LCD screen, b/w LCD screen, small alphanumeric LCD display or small numeric only

display), the available memory space and the available memory medium (RAM, hard disk or the like) or the system software available on the computer (JAVA and derivatives thereof, Windows, Windows-CE, etc.).

[0028] The individual adjustment of the agent systems installed on the respective computer means that high powered computers can have agent systems with a correspondingly high scope of performance installed on them, and lower powered computers can have simplified agent systems installed on them which are specifically tailored to the available software and hardware, including any peripheral units which are available.

[0029] Another advantage of the agent system is automatically adjusted to the respective environment or the computer and its installed software, so that nobody need worry about the correct configurations of the agent systems on the various computers in a network, even if it is not yet known from the outset which terminals are present on the network. This means that an environment can automatically develop in a network. The individual computers in the network need merely have agent system launchers installed on them.

[0030] In a network in which a full agent system has been installed on a single computer, it is sufficient, in principle, for only the agent system launcher to be installed on other computers onto which such an agent system is intended to be loaded, said agent system launcher then requesting, loading and starting an agent system adjusted to its environment.

[0031] Another advantage of the agent system is that the agent system launcher can be provided with an update function which tests whether a host computer is able to request a more recent version of the agent system and, if this is possible, the more up to date version of the agent system is requested, loaded on the client computer and started. Since such updates can generally rarely be performed, it is expedient for a full new agent system to be loaded in each case during an update. With very extensive agent systems which are regularly modified, it can be useful to use the update function to test whether individual parts of the agent system on the host computer are more up to date than those on the client computer, in which case possibly only individual parts of the agent system are then loaded from the host computer onto the client computer.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] These and other objects and advantages of the present invention will become more

apparent and more readily appreciated from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings of which:

[0033] Fig. 1 shows a network by which a plurality of computers are connected together and on which a respective agent system has been installed,

[0034] Fig.2 shows a block diagram of the design of an agent system, and

[0035] Fig.3 shows a flowchart for the method for loading an agent system from a host computer onto a client computer in a computer network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0036] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to like elements throughout.

[0037] Fig. 1 is a simplified schematic illustration of a network having a plurality of computers 1 which are connected to one another by data lines 2. The computers 1 have different scopes of performance, that is to say that they can have different processor powers, storage capacities and/or system programs. The power of the individual computers 1 is shown symbolically by the size of the rectangle representing the respective computer 1. The computers 1 are numbered from I. to IV., where the computer I. can be the most powerful and the computer IV. is a mobile terminal. The data line 2 from the computer IV. to the other computers is a mobile radio link which is not maintained permanently. It is therefore shown using a dashed line in Fig. 1.

[0038] The individual computers 1 have a respective agent system AS-I. to AS-IV. installed on them. The network contains a few mobile agents AG-I. to AG-IV., which are either arranged on one of the computers 1 or move from one computer to another (III. → II.).

[0039] Fig. 2 is a schematic illustration of the design of an agent system. The agent system has an agent platform 3 (also called platform) which comprises the service programs required by a mobile agent in order to be able to be executed on the respective computer 1. The agent platform 3 has two central program parts, the agent manager 4 and the communication manager 5. The communication manager 5 comprises a blackboard 6 and a plurality of protocol managers 7 providing the protocols for communication between agent systems and their applications, the agents 25. In addition, the communication manager allows the agents to

migrate. A network coupling 8 is used by the protocol managers 7 to allow the appropriate use of a connected network 9. The blackboard 6 can be used to buffer store data for the platform local communication between agents.

[0040] The agent manager 4 has a monitoring module 10 which is used to monitor internal operations for applications 25 connected to the agent manager 4. The monitoring information is processed by service programs, such as an agent software update program 11, a user manager 12, a security manager 13 and a resource manager 14. The agent manager 4 is also connected to a read only memory 15, for example represented by a hard disk. The read only memory 15 is used to store configuration data and important runtime information, such as agents, so that these are available again even after temporarily switching off. In small mobile computers IV, it is also customary for this read only memory 15 to be in the form of a semiconductor memory which can store data permanently, i.e. even after the computer has been switched off.

[0041] The agent manager 4 is provided with an administration interface 19 and an agent application interface 20. The administration interface 19 is used to connect the agent manager 4 to an agent system controller 21 which has an administration program 22 and an agent system launcher 23a and also an agent system update program 23b. The agent system launcher 23a is a service program which assists a user when starting and installing an agent system. The agent system update program 23b is a service program which can take particular presets as a basis for compiling an agent system and for sending it over the network 9. The agent system update program 23b is generally part of an existing agent system. However, it can also be installed on a computer independently of an agent system.

[0042] The agent application interface 20 is used to connect the agent manager 4 to an area of application 24 containing agents 25 which can execute particular applications. The agents 25 can reach the area of application 24 via the network 9, the communication manager 5 and the agent manager 4. Alternatively, a user of the computer on which the available agent system has been installed can load them into the area of application 24 using an agent launcher 26, and they can be activated in said area of application. The agent launcher 26 is a service program which assists a user in creating and activating agents 25.

[0043] The administration program 22 of the agent system controller 21 assists users in configuring and diagnosing the agent system.

[0044] The agent system launcher 23a of the agent system controller 21 is used for starting and loading the agent system onto a computer in the network. The operations performed in this context are shown in the flowchart shown in Fig. 3.

[0045] In order for this method for loading an agent system onto a computer to be possible, the computer onto which the agent system is being loaded and which is referred to below as the client computer 28 needs to have the agent system launcher 23a loaded on it. The client computer 28 needs to be connected by a data line 2 to at least one computer on which an agent system update program 23b has been installed, so that this computer, which is referred to below as the host computer 29, makes an adjusted or scaled copy of an agent system which is likewise available on it and can load it onto the client computer 28. The data line 2 can be an electronic or optical cable or a radio link.

[0046] The method for loading and updating an agent system is carried out by the agent system launcher 23a and is started in a step S1 (Fig. 3). This is followed by a test to determine whether an agent system AS already exists on the computer (S2). If this test reveals that no agent system exists on the present computer yet, program execution passes to step S3, which is part of a loading function 30 associated with the agent system launcher 23a. In step S3, an identifier identifying the hardware and/or software of the client computer 28 is sent to the host computer 29. This identifier contains statements relating to the power of the hardware and/or the software, such as the size of the available memory, the type of storage medium, the type of display and of the available system programs.

[0047] On the host computer, the identifier sent by the client computer 28 is assessed in a step S4, and an agent system is compiled in a step S5 on the basis of the identifier. The agent system is adjusted by adding or omitting components which are suitable/required or not required and configuring and scaling the selected components for the hardware and software available on the client computer 28. In addition, the components are configured in terms of their dynamic complexity, which is essentially based on the size of data structures. Besides the agent manager and the communication manager, the scalable service programs, such as the user manager 12, the security manager 13 and the resource manager 14, are adjusted, that is to say are scaled to the client computer 28. Steps S4 and S5 are performed in the agent system update program 23b arranged in the agent system controller 21 on the host computer 29.

[0048] When an agent system adjusted to the client computer 28 has been configured, this adjusted agent system is sent from the host computer 29 to the client computer in step S6. On the client computer 28, the agent system is received and loaded onto the client computer 28 in step S7. Step S7 is part of the loading function 30 of the agent system launcher 23. When the agent system has been loaded from the client computer, it is started in step S8, and the method for loading the agent system is then terminated in the subsequent step S9.

[0049] When the agent system is started in step S8, a starting and initialization routine (step S9) is called and is used to start and initialize the agent system. A startup sequence S10 is used to load stored agents (step S11). "Normal" operation of the agent system can then be executed, which is denoted by S12 in Fig. 3. If operation of the agent system needs to be terminated, the agents and runtime information for the agent system are first stored with a shutdown sequence (step S13). The agents are then terminated and the agent system is terminated (step S14).

[0050] If, on the other hand, step S2 described above establishes that an agent system already exists on the client computer 28, program execution passes to step S15, which tests whether updating of the agent system is permitted. If updating of the agent system is not permitted, program execution passes to step S16, which tests whether the agent system is in operation. If the agent system is not in operation, it is started in step S8 and the agent system launcher 23a is terminated in step S9. If the agent system is already in operation, program execution passes directly to step S9, where the agent system launcher 23a is terminated.

[0051] If the test in step S15 reveals that updating of the agent system is permitted, program execution passes to step S17, which tests whether the agent system is in operation.

[0052] If the agent system is in operation, step S18 calls step S13 (shutdown sequence) in order to terminate the operation of the agent system. A subsequent program loop (step S19) is used to test whether operation of the agent system has already ended. Only if the operation of the agent system has ended is program execution transferred to a step S20.

[0053] If the test in step S17 reveals that the agent system is not in operation, program execution can pass directly to step S20.

[0054] With step S20, a version identifier is sent to the host computer 29. The version identifier contains the information relating to the versions of the agent system, and components thereof,

installed on the client computer 28.

[0055] On the host computer 29, the version identifiers are evaluated in step S21 and, in the next step S22, version information for the most recent agent system, and components thereof, which can be loaded from the host computer are sent to the client computer. In step S23, the client computer 28 then compares its own version identifiers with the version information and of the host and tests whether the respective version which can be loaded from the host computer is more up to date than the version of the agent system, and components thereof, which is on the client computer. If the test reveals that the version which can be loaded from the host computer 29 is more up to date, program execution is transferred to step S3, whereupon an identifier identifying the hardware and/or software of the client computer 28 is in turn sent to the host computer, and the latter then compiles an agent system adjusted thereto and sends it to the client computer 28. If, on the other hand, the test in step S23 reveals that the versions which can be loaded from the host computer are not more up to date than the versions of the agent system, and components thereof, which are on the client computer, program execution is transferred to step S8, where the agent system is started. Hence, it is possible to update (during operation) the entire agent system or, if required, merely parts thereof.

[0056] Steps S20 and S23 form an update function 31 which the agent system launcher 23a uses to update an existing agent system.

[0057] The method for loading an agent system is carried out by the agent system launcher 23a. The agent system launcher can be started manually at any time, irrespective of whether or not an agent system is available or whether or not the agent system is in operation. The agent system launcher 23a can also be activated automatically when it is first loaded onto a computer or when a computer is started. If a full agent system already exists, then the agent system launcher 23a can be controlled directly by the user or by the administration program 22, with the administration program 22 being able to call it repeatedly, thus ensuring that the agent system is updated regularly and according to requirements.

[0058] To manage the agent systems on a plurality of mobile client computers 28, a host computer 29 permanently connected to a network can be provided. If, by way of example, a small mobile computer, the client computer 28, having a black and white screen is to start an agent system, then the host computer 29 sends a request to the client computer 28 and in so

doing also transfers the current data relating to its resources (e.g. b/w screen, 8 MB RAM, maximum number of windows which can be shown on the screen, Java Virtual Machine available, graphical display of characters, user management functionality in the agent system for one or more users, etc.). The host computer 29 uses this information to compile an agent system which contains, by way of example, a Java class for printing specifically on b/w appliances. The equivalent class for printing on appliances with a color screen or a screen which can show only ASCII characters would have a different appearance. Depending on a Java Virtual Machine (Persona Java, JDK1.x, 1.2, etc.) available on the client computer 28, the appropriate classes are compiled automatically. If only one user needs to be managed, as compared with a plurality, the appropriate class providing the user management is also correspondingly smaller and has a smaller resource requirement on the client computer 28.

[0059] The agent system can optionally be provided with a function which automatically tests the hardware and/or software of the respective computer and starts the launcher 23a if required. This means that the agent system can be adjusted dynamically to changes in the hardware and/or in the software. This function can be provided by step S24 in Fig. 3, which follows step S17. Step S24 tests whether the hardware and/or software has been changed since the agent system was last updated. If such a change is established, program execution passes directly to step S3, where an appropriate identifier is sent to the host computer 29. If such a change is not established, the program passes to step S20 and is executed in line with the description above. The programs on which the agent systems are based can be stored and sold on electronically readable data stores.

[0060] The invention has been described in detail with particular reference to preferred embodiments thereof and examples, but it will be understood that variations and modifications can be effected within the spirit and scope of the invention.

MARKED-UP TRANSLATION OF INTERNATIONAL APPLICATION

[Description]TITLE OF THE INVENTION

AGENT SYSTEM <u>FOR MOBILE AGENTS</u>, COMPUTER, NETWORK AND METHOD FOR <u>DOWN</u>LOADING AN AGENT SYSTEM FROM A HOST COMPUTER [ONTO]<u>TO</u> A CLIENT COMPUTER [IN]<u>OF</u> A COMPUTER NETWORK

<u>CROSS REFERENCE TO RELATED APPLICATIONS</u>

[0001] <u>This application is based on and hereby claims priority to PCT Application No. PCT/DE00/02705 filed on August 11, 2000, German Patent Application No. 199 39 058.4 filed on August 18, 1999, the contents of which are hereby incorporated by reference.</u>

<u>BACKGROUND OF THE INVENTION</u>

[0002] The invention relates to an agent system, in particular an agent system for mobile agents, to a computer network having a plurality of computers networked by [means of] data [line]<u>lines</u> and to a method for loading an agent system from a host computer onto a client computer.

[0003] Agents are autonomous, cooperative software units comprising code and data. They are independently operating software units which require no continual interaction with the user. There are both static and mobile agents.

[0004] Mobile agents are known, by way of example, from US 5,603,031 and from the book Software Agents, Bradshaw Jeffrey, Menlo Park, California: AAAI Press/The MIT Press, 1996. The chapter on mobile agents in this book can be retrieved in a slightly modified version on the Internet at the address http://www.generalmagic.com/technology/techwhitepaper.html.

[0005] Mobile agents are programs which can be active at various stations in a computer network and can change their location in the computer network. Conventionally, programs in a network are loaded from one computer onto another computer while they are not active. By contrast, mobile agents are first activated and automatically change their position in the network while they are active.

[0006] This small but fundamental difference with respect to conventional programs has considerable effects on the type of applications on the mobile agents in the network. The agents are created at a particular station in the network, which involves stipulating the functions needing to be performed by the agent, so that the agent, having been transmitted to another

station in the network, represents the previous station in the network and performs the appropriate functions. This means that functions can be stipulated at one station in the network which are performed at another station in the network. In principle, this is also possible with conventional systems, with this requiring a permanent exchange of information in the network between the two stations so that the station at which the appropriate control operations are stipulated transmits them to the station at which they are performed.

[0007] By contrast, the mobile agents require no such control operations, since their functions are stipulated once in advance and are then performed automatically. This significantly reduces the data throughput in the network, with any desired operations being able to be performed at the stations remote from the task issuer.

[0008] However, the agents do not always need to operate fully detached from their initiator, but rather can set up communication channels thereto or also to other stations in the network.

[0009] When moving from one station in a network to another station in the network, the mobile agents execute a migration command (e.g. "go"). Such a command requires a destination description containing the name, address or type of the station to which the agent is intended to move, or a combination of these. Different stations do not always need to be arranged at different computers in the network. It is also possible for agents to be active at different stations within one computer.

[0010] Mobile agents can also multiply, so that they perform their functions independently of one another at different stations.

[0011] Mobile agents are frequently used for collecting predetermined data, some of which are evaluated in situ, so that an already preprocessed result is sent to the original task issuer.

[0012] One possible application for such mobile agents arises, by way of example, when a purchaser of a particular object receives a low price guarantee from the vendor. The purchaser can then send out one or more mobile agents which independently obtain corresponding offers from other suppliers, possibly monitor them over a certain period of time and send them to the purchaser, so that he can compare them with the price offered by the vendor.

[0013] Other, known mobile agents and agent systems are marketed under the names Aglets, Voyager and Concordia and can generally be obtained over the Internet.

[0014] So that the mobile agents on the individual computers in the network can be active, these computers need to have a respective "agent system". An agent system is a collection of service programs which need the respective agents for their tasks. These service programs generally comprise an agent manager connected to other service programs, so that the active mobile agents can fall back on the respective computer resources and can migrate from one agent system in the network to another agent system in the network via network interfaces. Generally, an agent system controller is provided which regulates the execution of the agent system on the respective computer.

[0015] These agent systems form the stations in the network at which the agents can become active.

[0016] Such agent systems are generally written in a platform-independent computer language, such as JAVA, so that they can run on any computer systems.

[0017] The transfer of the agents in the network while they are active creates a new programming paradigm according to which the program code itself is moved to the data needing to be processed instead of data being transported to the processing code or the processing code needing to be loaded in the inactive state onto the respective computer.

[0018] Agents are generally relatively small programs which are sent out and exploit the intelligence available in situ. Since mobile agents can also perform their tasks when there is no permanent network connection between the individual computers, they are ideally suited to use on mobile computers, which are generally not permanently connected to a network.

[0019] However, the agent systems known to date are too extensive for them to be able to be installed on arbitrary simple mobile computers.

[0020] Within the context of the present patent application, the term computer also covers simple process computers equipped merely with a minimal hardware configuration (processor, memory and display). Such simple process computers are integrated on a single chip, for example.

SUMMARY OF THE INVENTION

[0021] [The]One possible object of the invention is [therefore based on the object of providing]to provide an agent system which is able to offer the scope of functions of known agent systems

and is nevertheless also suitable for small computers, such as small process computers.

[0022] [The]Another possible object of the invention is [also based on the object of providing]to provide an appropriate computer network and a method for loading an agent system from a host computer onto a client computer in a computer network. [The objects are achieved by an agent system having the features of claim 1, by a computer network having the features of claim 6 and by a method having the features of claim 9. Advantageous refinements of the invention are specified in the subclaims.]

[0023] The [inventive] agent system, which can be installed on computers connected in a computer network, has an agent platform and an agent system controller. The agent platform comprises service programs required by a mobile agent in order to be able to be executed on the respective computer. The agent system controller contains an agent system launcher which can be loaded onto a client computer independently of the rest of the agent system and is used for loading the agent system from a host computer.

[0024] The [inventive] agent system is distinguished in that the agent system launcher is designed such that, before an agent system is loaded from a host computer onto the client computer on which at least the agent system launcher is installed, an identifier describing the hardware and/or software of the client computer is sent to the host computer, and a server system, the agent system update program, installed on the host computer is designed such that it takes the identifier describing the hardware and/or the software of the client computer as a basis for compiling an agent system adjusted to the hardware and/or software of the client computer and loads this agent system onto the client computer.

[0025] Within the context of the present invention, a host computer is any computer in the network from which it is possible to load an agent system onto another computer, which is referred to as a client computer[within the context of the present invention]. It is thus possible for any computer in the network to be a host computer and a client computer, irrespective of its function otherwise in the network.

[0026] The [inventive refinement of the] agent system launcher, which sends an identifier identifying the client computer to the host computer, and the [inventive refinement of the] agent system which is installed on the host computer and takes this identifier as a basis for compiling an adjusted agent system and makes this adjusted agent system available to the client

computer make it possible for even small, in particular mobile, computers, for example comprising a simple process computer or a personal digital assistant, to be incorporated into a computer network and to be provided with the [inventive] agent system, since this agent system is automatically adjusted to the respective client computer and its power.

[0027] Typical criteria for adjustment are, by way of example, the display (e.g. color picture tube, color LCD screen, b/w LCD screen, small alphanumeric LCD display or small numeric only display), the available memory space and the available memory medium (RAM, hard disk or the like) or the system software available on the computer (JAVA and derivatives thereof, Windows, Windows-CE, etc.).

[0028] The individual adjustment of the agent systems installed on the respective computer means that high powered computers can have agent systems with a correspondingly high scope of performance installed on them, and lower powered computers can have simplified agent systems installed on them which are specifically tailored to the available software and hardware, including any peripheral units which are available.

[0029] Another advantage of the [invention is that the] agent system is automatically adjusted to the respective environment or the computer and its installed software, so that nobody need worry about the correct configurations of the agent systems on the various computers in a network, even if it is not yet known from the outset which terminals are present on the network. This means that an environment can automatically develop in a network. The individual computers in the network need merely have agent system launchers installed on them.

[0030] In a network in which a full agent system has been installed on a single computer, it is sufficient, in principle, for only the agent system launcher to be installed on other computers onto which such an agent system is intended to be loaded, said agent system launcher then requesting, loading and starting an agent system adjusted to its environment.

[0031] Another advantage of the [inventive] agent system is that the agent system launcher can be provided with an update function which tests whether a host computer is able to request a more recent version of the agent system and, if this is possible, the more up to date version of the agent system is requested, loaded on the client computer and started. Since such updates can generally rarely be performed, it is expedient for a full new agent system to be loaded in each case during an update. With very extensive agent systems which are regularly modified, it

can be useful to use the update function to test whether individual parts of the agent system on the host computer are more up to date than those on the client computer, in which case possibly only individual parts of the agent system are then loaded from the host computer onto the client computer.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] [The invention is explained in more detail below using an exemplary embodiment shown in the drawings, in which, schematically:]These and other objects and advantages of the present invention will become more apparent and more readily appreciated from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings of which:

[0033] [figure]Fig. 1 shows a network by [means of] which a plurality of computers are connected together and on which a respective agent system has been installed,

[0034] [figure]Fig.2 shows a block diagram of the design of an agent system, and

[0035] [figure]Fig.3 shows a flowchart for the method for loading an agent system from a host computer onto a client computer in a computer network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0036] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to like elements throughout.

[0037] [Figure]Fig. 1 is a simplified schematic illustration of a network having a plurality of computers 1 which are connected to one another by [means of] data lines 2. The computers 1 have different scopes of performance, that is to say that they can have different processor powers, storage capacities and/or system programs. The power of the individual computers 1 is shown symbolically by the size of the rectangle representing the respective computer 1. The computers 1 are numbered from I. to IV., where the computer I. can be the most powerful and the computer IV. is a mobile terminal. The data line 2 from the computer IV. to the other computers is a mobile radio link which is not maintained permanently. It is therefore shown using a dashed line in [figure]Fig. 1.

[0038] The individual computers 1 have a respective agent system AS-I. to AS-IV. installed on

them. The network contains a few mobile agents AG-I. to AG-IV., which are either arranged on one of the computers 1 or move from one computer to another (III. → II.).

[0039] [Figure]Fig. 2 is a schematic illustration of the design of an agent system. The agent system has an agent platform 3 (also called platform) which comprises the service programs required by a mobile agent in order to be able to be executed on the respective computer 1. The agent platform 3 has two central program parts, the agent manager 4 and the communication manager 5. The communication manager 5 comprises a blackboard 6 and a plurality of protocol managers 7 providing the protocols for communication between agent systems and their applications, the agents 25. In addition, the communication manager allows the agents to migrate. A network coupling 8 is used by the protocol managers 7 to allow the appropriate use of a connected network 9. The blackboard 6 can be used to buffer store data for the platform local communication between agents.

[0040] The agent manager 4 has a monitoring module 10 which is used to monitor internal operations for applications 25 connected to the agent manager 4. The monitoring information is processed by service programs, such as an agent software update program 11, a user manager 12, a security manager 13 and a resource manager 14. The agent manager 4 is also connected to a read only memory 15, for example represented by a hard disk. The read only memory 15 is used to store configuration data and important runtime information, such as agents, so that these are available again even after temporarily switching off. In small mobile computers IV, it is also customary for this read only memory 15 to be in the form of a semiconductor memory which can store data permanently, i.e. even after the computer has been switched off.

[0041] The agent manager 4 is provided with an administration interface 19 and an agent application interface 20. The administration interface 19 is used to connect the agent manager 4 to an agent system controller 21 which has an administration program 22 and an agent system launcher 23a and also an agent system update program 23b. The agent system launcher 23a is a service program which assists a user when starting and installing an agent system. The agent system update program 23b is a service program which can take particular presets as a basis for compiling an agent system and for sending it over the network 9. The agent system update program 23b is generally part of an existing agent system. However, it can also be installed on a computer independently of an agent system.

[0042] The agent application interface 20 is used to connect the agent manager 4 to an area of application 24 containing agents 25 which can execute particular applications. The agents 25 can reach the area of application 24 via the network 9, the communication manager 5 and the agent manager 4. Alternatively, a user of the computer on which the available agent system has been installed can load them into the area of application 24 using an agent launcher 26, and they can be activated in said area of application. The agent launcher 26 is a service program which assists a user in creating and activating agents 25.

[0043] The administration program 22 of the agent system controller 21 assists users in configuring and diagnosing the agent system.

[0044] The agent system launcher 23a of the agent system controller 21 is used for starting and loading the agent system onto a computer in the network. The operations performed in this context are shown in the flowchart shown in [figure]Fig. 3.

[0045] In order for this method for loading an agent system onto a computer to be possible, the computer onto which the agent system is being loaded and which is referred to below as the client computer 28 needs to have the agent system launcher 23a loaded on it. The client computer 28 needs to be connected by [means of] a data line 2 to at least one computer on which an agent system update program 23b has been installed, so that this computer, which is referred to below as the host computer 29, makes an adjusted or scaled copy of an agent system which is likewise available on it and can load it onto the client computer 28. The data line 2 can be an electronic or optical cable or a radio link.

[0046] The method for loading and updating an agent system is carried out by the agent system launcher 23a and is started in a step S1 ([figure]Fig. 3). This is followed by a test to determine whether an agent system AS already exists on the computer (S2). If this test reveals that no agent system exists on the present computer yet, program execution passes to step S3, which is part of a loading function 30 associated with the agent system launcher 23a. In step S3, an identifier identifying the hardware and/or software of the client computer 28 is sent to the host computer 29. This identifier contains statements relating to the power of the hardware and/or the software, such as the size of the available memory, the type of storage medium, the type of display and of the available system programs.

[0047] On the host computer, the identifier sent by the client computer 28 is assessed in a step

S4, and an agent system is compiled in a step S5 on the basis of the identifier. The agent system is adjusted by adding or omitting components which are suitable/required or not required and configuring and scaling the selected components for the hardware and software available on the client computer 28. In addition, the components are configured in terms of their dynamic complexity, which is essentially based on the size of data structures. Besides the agent manager and the communication manager, the scalable service programs, such as the user manager 12, the security manager 13 and the resource manager 14, are adjusted, that is to say are scaled to the client computer 28. Steps S4 and S5 are performed in the agent system update program 23b arranged in the agent system controller 21 on the host computer 29.

[0048] When an agent system adjusted to the client computer 28 has been configured, this adjusted agent system is sent from the host computer 29 to the client computer in step S6. On the client computer 28, the agent system is received and loaded onto the client computer 28 in step S7. Step S7 is part of the loading function 30 of the agent system launcher 23. When the agent system has been loaded from the client computer, it is started in step S8, and the method for loading the agent system is then terminated in the subsequent step S9.

[0049] When the agent system is started in step S8, a starting and initialization routine (step S9) is called and is used to start and initialize the agent system. A startup sequence S10 is used to load stored agents (step S11). "Normal" operation of the agent system can then be executed, which is denoted by S12 in [figure]Fig. 3. If operation of the agent system needs to be terminated, the agents and runtime information for the agent system are first stored with a shutdown sequence (step S13). The agents are then terminated and the agent system is terminated (step S14).

[0050] If, on the other hand, step S2 described above establishes that an agent system already exists on the client computer 28, program execution passes to step S15, which tests whether updating of the agent system is permitted. If updating of the agent system is not permitted, program execution passes to step S16, which tests whether the agent system is in operation. If the agent system is not in operation, it is started in step S8 and the agent system launcher 23a is terminated in step S9. If the agent system is already in operation, program execution passes directly to step S9, where the agent system launcher 23a is terminated.

[0051] If the test in step S15 reveals that updating of the agent system is permitted, program

execution passes to step S17, which tests whether the agent system is in operation.

[0052] If the agent system is in operation, step S18 calls step S13 (shutdown sequence) in order to terminate the operation of the agent system. A subsequent program loop (step S19) is used to test whether operation of the agent system has already ended. Only if the operation of the agent system has ended is program execution transferred to a step S20.

[0053] If the test in step S17 reveals that the agent system is not in operation, program execution can pass directly to step S20.

[0054] With step S20, a version identifier is sent to the host computer 29. The version identifier contains the information relating to the versions of the agent system, and components thereof, installed on the client computer 28.

[0055] On the host computer 29, the version identifiers are evaluated in step S21 and, in the next step S22, version information for the most recent agent system, and components thereof, which can be loaded from the host computer are sent to the client computer. In step S23, the client computer 28 then compares its own version identifiers with the version information and [lacuna] of the host and tests whether the respective version which can be loaded from the host computer is more up to date than the version of the agent system, and components thereof, which is on the client computer. If the test reveals that the version which can be loaded from the host computer 29 is more up to date, program execution is transferred to step S3, whereupon an identifier identifying the hardware and/or software of the client computer 28 is in turn sent to the host computer, and the latter then compiles an agent system adjusted thereto and sends it to the client computer 28. If, on the other hand, the test in step S23 reveals that the versions which can be loaded from the host computer are not more up to date than the versions of the agent system, and components thereof, which are on the client computer, program execution is transferred to step S8, where the agent system is started. Hence, it is possible to update (during operation) the entire agent system or, if required, merely parts thereof.

[0056] Steps S20 and S23 form an update function 31 which the agent system launcher 23a uses to update an existing agent system.

[0057] The [inventive] method for loading an agent system is carried out by the agent system launcher 23a. The agent system launcher can be started manually at any time, irrespective of whether or not an agent system is available or whether or not the agent system is in operation.

The agent system launcher 23a can also be activated automatically when it is first loaded onto a computer or when a computer is started. If a full agent system already exists, then the agent system launcher 23a can be controlled directly by the user or by the administration program 22, with the administration program 22 being able to call it repeatedly, thus ensuring that the agent system is updated regularly and according to requirements.

[0058] To manage the agent systems on a plurality of mobile client computers 28, a host computer 29 permanently connected to a network can be provided. If, by way of example, a small mobile computer, the client computer 28, having a black and white screen is to start an agent system, then the host computer 29 sends a request to the client computer 28 and in so doing also transfers the current data relating to its resources (e.g. b/w screen, 8 MB RAM, maximum number of windows which can be shown on the screen, Java Virtual Machine available, graphical display of characters, user management functionality in the agent system for one or more users, etc.). The host computer 29 uses this information to compile an agent system which contains, by way of example, a Java class for printing specifically on b/w appliances. The equivalent class for printing on appliances with a color screen or a screen which can show only ASCII characters would have a different appearance. Depending on a Java Virtual Machine (Persona Java, JDK1.x, 1.2, etc.) available on the client computer 28, the appropriate classes are compiled automatically. If only one user needs to be managed, as compared with a plurality, the appropriate class providing the user management is also correspondingly smaller and has a smaller resource requirement on the client computer 28.

[0059] The [inventive] agent system can optionally be provided with a function which automatically tests the hardware and/or software of the respective computer and starts the launcher 23a if required. This means that the agent system can be adjusted dynamically to changes in the hardware and/or in the software. This function can be provided by step S24 in [figure]Fig. 3, which follows step S17. Step S24 tests whether the hardware and/or software has been changed since the agent system was last updated. If such a change is established, program execution passes directly to step S3, where an appropriate identifier is sent to the host computer 29. If such a change is not established, the program passes to step S20 and is executed in line with the description above. The programs on which the [inventive] agent systems are based can be stored and sold on electronically readable data stores.

[0060] The invention has been described in detail with particular reference to preferred

embodiments thereof and examples, but it will be understood that variations and modifications can be effected within the spirit and scope of the invention.

3/Prtx

GR 99 P 2597

Description

Agent system, computer network and method for loading
an agent system from a host computer onto a client
computer in a computer network

The invention relates to an agent system, in particular
an agent system for mobile agents, to a computer
network having a plurality of computers networked by
means of data line and to a method for loading an agent
system from a host computer onto a client computer.

Agents are autonomous, cooperative software units
comprising code and data. They are independently
operating software units which require no continual
interaction with the user. There are both static and
mobile agents.

Mobile agents are known, by way of example, from
US 5,603,031 and from the book Software Agents,
Bradshaw Jeffrey, Menlo Park, California: AAAI
Press/The MIT Press, 1996. The chapter on mobile agents
in this book can be retrieved in a slightly modified
version on the Internet at the address
http://www.generalmagic.com/technology/techwhitepaper.html.

Mobile agents are programs which can be active at
various stations in a computer network and can change
their location in the computer network. Conventionally,
programs in a network are loaded from one computer onto
another computer while they are not active. By
contrast, mobile agents are first activated and
automatically change their position in the network
while they are active.

This small but fundamental difference with respect to
conventional programs has considerable effects on the

type of applications on the mobile agents in the
network. The agents are

created at a particular station in the network, which involves stipulating the functions needing to be performed by the agent, so that the agent, having been transmitted to another station in the network,

5 represents the previous station in the network and performs the appropriate functions. This means that functions can be stipulated at one station in the network which are performed at another station in the network. In principle, this is also possible with

10 conventional systems, with this requiring a permanent exchange of information in the network between the two stations so that the station at which the appropriate control operations are stipulated transmits them to the station at which they are performed.

15

By contrast, the mobile agents require no such control operations, since their functions are stipulated once in advance and are then performed automatically. This significantly reduces the data throughput in the

20 network, with any desired operations being able to be performed at the stations remote from the task issuer.

However, the agents do not always need to operate fully detached from their initiator, but rather can set up

25 communication channels thereto or also to other stations in the network.

When moving from one station in a network to another station in the network, the mobile agents execute a

30 migration command (e.g. "go"). Such a command requires a destination description containing the name, address or type of the station to which the agent is intended to move, or a combination of these. Different stations do not always need to be arranged at different

35 computers in the network. It is also possible for

agents to be active at different stations within one computer.

Mobile agents can also multiply, so that they perform their functions independently of one another at different stations.

5   Mobile agents are frequently used for collecting predetermined data, some of which are evaluated in situ, so that an already preprocessed result is sent to the original task issuer.

10   One possible application for such mobile agents arises, by way of example, when a purchaser of a particular object receives a low price guarantee from the vendor. The purchaser can then send out one or more mobile agents which independently obtain corresponding offers

15   from other suppliers, possibly monitor them over a certain period of time and send them to the purchaser, so that he can compare them with the price offered by the vendor.

20   Other, known mobile agents and agent systems are marketed under the names Aglets, Voyager and Concordia and can generally be obtained over the Internet.

So that the mobile agents on the individual computers

25   in the network can be active, these computers need to have a respective "agent system". An agent system is a collection of service programs which need the respective agents for their tasks. These service programs generally comprise an agent manager connected

30   to other service programs, so that the active mobile agents can fall back on the respective computer resources and can migrate from one agent system in the network to another agent system in the network via network interfaces. Generally, an agent system

35   controller is provided which regulates the execution of the agent system on the respective computer.

These agent systems form the stations in the network at which the agents can become active.

Such agent systems are generally written in a platform-independent computer language, such as JAVA, so that they can run on any computer systems.

The transfer of the agents in the network while they are active creates a new programming paradigm according to which the program code itself is moved to the data needing to be processed instead of data being transported to the processing code or the processing code needing to be loaded in the inactive state onto the respective computer.

Agents are generally relatively small programs which are sent out and exploit the intelligence available in situ. Since mobile agents can also perform their tasks when there is no permanent network connection between the individual computers, they are ideally suited to use on mobile computers, which are generally not permanently connected to a network.

However, the agent systems known to date are too extensive for them to be able to be installed on arbitrary simple mobile computers.

Within the context of the present patent application, the term computer also covers simple process computers equipped merely with a minimal hardware configuration (processor, memory and display). Such simple process computers are integrated on a single chip, for example.

The invention is therefore based on the object of providing an agent system which is able to offer the scope of functions of known agent systems and is nevertheless also suitable for small computers, such as small process computers. The

invention is also based on the object of providing an appropriate computer network and a method for loading an agent system from a host computer onto a client computer in a computer network. The objects are

5 achieved by an agent system having the features of claim 1, by a computer network having the features of claim 6 and by a method having the features of claim 9. Advantageous refinements of the invention are specified in the subclaims.

10

The inventive agent system, which can be installed on computers connected in a computer network, has an agent platform and an agent system controller. The agent platform comprises service programs required by a

15 mobile agent in order to be able to be executed on the respective computer. The agent system controller contains an agent system launcher which can be loaded onto a client computer independently of the rest of the agent system and is used for loading the agent system

20 from a host computer.

The inventive agent system is distinguished in that the agent system launcher is designed such that, before an agent system is loaded from a host computer onto the

25 client computer on which at least the agent system launcher is installed, an identifier describing the hardware and/or software of the client computer is sent to the host computer, and a server system, the agent system update program, installed on the host computer

30 is designed such that it takes the identifier describing the hardware and/or the software of the client computer as a basis for compiling an agent system adjusted to the hardware and/or software of the client computer and loads this agent system onto the

35 client computer.

Within the context of the present invention, a host

computer is any computer in the network from which it
is possible to load an agent system onto another
computer, which is

referred to as a client computer within the context of the present invention. It is thus possible for any computer in the network to be a host computer and a client computer, irrespective of its function otherwise
5   in the network.

The inventive refinement of the agent system launcher, which sends an identifier identifying the client computer to the host computer, and the inventive
10 refinement of the agent system which is installed on the host computer and takes this identifier as a basis for compiling an adjusted agent system and makes this adjusted agent system available to the client computer make it possible for even small, in particular mobile,
15 computers, for example comprising a simple process computer or a personal digital assistant, to be incorporated into a computer network and to be provided with the inventive agent system, since this agent system is automatically adjusted to the respective
20 client computer and its power.

Typical criteria for adjustment are, by way of example, the display (e.g. color picture tube, color LCD screen, b/w LCD screen, small alphanumeric LCD display or small
25 numeric only display), the available memory space and the available memory medium (RAM, hard disk or the like) or the system software available on the computer (JAVA and derivatives thereof, Windows, Windows-CE, etc.).
30
The individual adjustment of the agent systems installed on the respective computer means that high powered computers can have agent systems with a correspondingly high scope of performance installed on
35 them, and lower powered computers can have simplified agent systems installed on them which are specifically tailored to the available software and hardware, including any peripheral units which are available.

Another advantage of the invention is that the agent
system is automatically adjusted to the respective
environment or the computer and its installed software,
so that nobody need worry about the correct

5   configurations of the agent systems on the various
computers in a network, even if it is not yet known
from the outset which terminals are present on the
network. This means that an environment can
automatically develop in a network. The individual

10 computers in the network need merely have agent system
launchers installed on them.

In a network in which a full agent system has been
installed on a single computer, it is sufficient, in

15 principle, for only the agent system launcher to be
installed on other computers onto which such an agent
system is intended to be loaded, said agent system
launcher then requesting, loading and starting an agent
system adjusted to its environment.

20
Another advantage of the inventive agent system is that
the agent system launcher can be provided with an
update function which tests whether a host computer is
able to request a more recent version of the agent

25 system and, if this is possible, the more up to date
version of the agent system is requested, loaded on the
client computer and started. Since such updates can
generally rarely be performed, it is expedient for a
full new agent system to be loaded in each case during

30 an update. With very extensive agent systems which are
regularly modified, it can be useful to use the update
function to test whether individual parts of the agent
system on the host computer are more up to date than
those on the client computer, in which case possibly

35 only individual parts of the agent system are then
loaded from the host computer onto the client computer.

The invention is explained in more detail below using an exemplary embodiment shown in the drawings, in which, schematically:

5   figure 1   shows a network by means of which a plurality of computers are connected together and on which a respective agent system has been installed,

10   figure 2   shows a block diagram of the design of an agent system, and

figure 3   shows a flowchart for the method for loading an agent system from a host computer onto a
15   client computer in a computer network.

Figure 1 is a simplified schematic illustration of a network having a plurality of computers 1 which are connected to one another by means of data lines 2. The
20   computers 1 have different scopes of performance, that is to say that they can have different processor powers, storage capacities and/or system programs. The power of the individual computers 1 is shown symbolically by the size of the rectangle representing
25   the respective computer 1. The computers 1 are numbered from I. to IV., where the computer I. can be the most powerful and the computer IV. is a mobile terminal. The data line 2 from the computer IV. to the other computers is a mobile radio link which is not
30   maintained permanently. It is therefore shown using a dashed line in figure 1.

The individual computers 1 have a respective agent system AS-I. to AS-IV. installed on them. The network
35   contains a few mobile agents AG-I. to AG-IV., which are either arranged on one of the computers 1 or move from one computer to another (III. → II.).

Figure 2 is a schematic illustration of the design of an agent system. The agent system has an agent platform 3 (also called platform) which comprises the service programs required by a mobile agent in order to be able

5 to be executed on the respective computer 1. The agent platform 3 has two central program parts, the agent manager 4 and the communication manager 5. The communication manager 5 comprises a blackboard 6 and a plurality of protocol managers 7 providing the

10 protocols for communication between agent systems and their applications, the agents 25. In addition, the communication manager allows the agents to migrate. A network coupling 8 is used by the protocol managers 7 to allow the appropriate use of a connected network 9.

15 The blackboard 6 can be used to buffer store data for the platform local communication between agents.

The agent manager 4 has a monitoring module 10 which is used to monitor internal operations for applications 25

20 connected to the agent manager 4. The monitoring information is processed by service programs, such as an agent software update program 11, a user manager 12, a security manager 13 and a resource manager 14. The agent manager 4 is also connected to a read only memory

25 15, for example represented by a hard disk. The read only memory 15 is used to store configuration data and important runtime information, such as agents, so that these are available again even after temporarily switching off. In small mobile computers IV, it is also

30 customary for this read only memory 15 to be in the form of a semiconductor memory which can store data permanently, i.e. even after the computer has been switched off.

35 The agent manager 4 is provided with an administration interface 19 and an agent application interface 20. The administration interface 19 is used to connect the agent manager 4 to an agent system controller 21

which has an administration program 22 and an agent
system launcher 23a and also an agent system update
program 23b. The agent system launcher 23a is a service
program which assists a user when starting and
5 installing an agent system. The agent system update
program 23b is a service program which can take
particular presets as a basis for compiling an agent
system and for sending it over the network 9. The agent
system update program 23b is generally part of an
10 existing agent system. However, it can also be
installed on a computer independently of an agent
system.

The agent application interface 20 is used to connect
15 the agent manager 4 to an area of application 24
containing agents 25 which can execute particular
applications. The agents 25 can reach the area of
application 24 via the network 9, the communication
manager 5 and the agent manager 4. Alternatively, a
20 user of the computer on which the available agent
system has been installed can load them into the area
of application 24 using an agent launcher 26, and they
can be activated in said area of application. The agent
launcher 26 is a service program which assists a user
25 in creating and activating agents 25.

The administration program 22 of the agent system
controller 21 assists users in configuring and
diagnosing the agent system.
30
The agent system launcher 23a of the agent system
controller 21 is used for starting and loading the
agent system onto a computer in the network. The
operations performed in this context are shown in the
35 flowchart shown in figure 3.

In order for this method for loading an agent system

onto a computer to be possible, the computer onto which
the agent system is being loaded and which is referred
to below as

the client computer 28 needs to have the agent system launcher 23a loaded on it. The client computer 28 needs to be connected by means of a data line 2 to at least one computer on which an agent system update program

5   23b has been installed, so that this computer, which is referred to below as the host computer 29, makes an adjusted or scaled copy of an agent system which is likewise available on it and can load it onto the client computer 28. The data line 2 can be an

10 electronic or optical cable or a radio link.

The method for loading and updating an agent system is carried out by the agent system launcher 23a and is started in a step S1 (figure 3). This is followed by a

15 test to determine whether an agent system AS already exists on the computer (S2). If this test reveals that no agent system exists on the present computer yet, program execution passes to step S3, which is part of a loading function 30 associated with the agent system

20 launcher 23a. In step S3, an identifier identifying the hardware and/or software of the client computer 28 is sent to the host computer 29. This identifier contains statements relating to the power of the hardware and/or the software, such as the size of the available memory,

25 the type of storage medium, the type of display and of the available system programs.

On the host computer, the identifier sent by the client computer 28 is assessed in a step S4, and an agent

30 system is compiled in a step S5 on the basis of the identifier. The agent system is adjusted by adding or omitting components which are suitable/required or not required and configuring and scaling the selected components for the hardware and software available on

35 the client computer 28. In addition, the components are configured in terms of their dynamic complexity, which

is essentially based on the size of data structures.
Besides the agent manager and the communication
manager,

the scalable service programs, such as the user manager
12, the security manager 13 and the resource manager
14, are adjusted, that is to say are scaled to the
client computer 28. Steps S4 and S5 are performed in
5   the agent system update program 23b arranged in the
agent system controller 21 on the host computer 29.

When an agent system adjusted to the client computer 28
has been configured, this adjusted agent system is sent
10 from the host computer 29 to the client computer in
step S6. On the client computer 28, the agent system is
received and loaded onto the client computer 28 in step
S7. Step S7 is part of the loading function 30 of the
agent system launcher 23. When the agent system has
15 been loaded from the client computer, it is started in
step S8, and the method for loading the agent system is
then terminated in the subsequent step S9.

When the agent system is started in step S8, a starting
20 and initialization routine (step S9) is called and is
used to start and initialize the agent system. A
startup sequence S10 is used to load stored agents
(step S11). "Normal" operation of the agent system can
then be executed, which is denoted by S12 in figure 3.
25 If operation of the agent system needs to be
terminated, the agents and runtime information for the
agent system are first stored with a shutdown sequence
(step S13). The agents are then terminated and the
agent system is terminated (step S14).
30
If, on the other hand, step S2 described above
establishes that an agent system already exists on the
client computer 28, program execution passes to step
S15, which tests whether updating of the agent system
35 is permitted. If updating of the agent system is not
permitted, program execution passes to step S16, which
tests whether the agent system is

in operation. If the agent system is not in operation,
it is started in step S8 and the agent system launcher
23a is terminated in step S9. If the agent system is
already in operation, program execution passes directly
5   to step S9, where the agent system launcher 23a is
terminated.

If the test in step S15 reveals that updating of the
agent system is permitted, program execution passes to
10 step S17, which tests whether the agent system is in
operation.

If the agent system is in operation, step S18 calls
step S13 (shutdown sequence) in order to terminate the
15 operation of the agent system. A subsequent program
loop (step S19) is used to test whether operation of
the agent system has already ended. Only if the
operation of the agent system has ended is program
execution transferred to a step S20.
20
If the test in step S17 reveals that the agent system
is not in operation, program execution can pass
directly to step S20.

25 With step S20, a version identifier is sent to the host
computer 29. The version identifier contains the
information relating to the versions of the agent
system, and components thereof, installed on the client
computer 28.
30
On the host computer 29, the version identifiers are
evaluated in step S21 and, in the next step S22,
version information for the most recent agent system,
and components thereof, which can be loaded from the
35 host computer are sent to the client computer. In step
S23, the client computer 28 then compares its own
version identifiers with the version information and

[lacuna] of the host and tests whether the respective version which can be loaded from the host computer is more up to date than the version of the agent system, and components thereof, which is on the

5

client computer. If the test reveals that the version which can be loaded from the host computer 29 is more up to date, program execution is transferred to step S3, whereupon an identifier identifying the hardware
5   and/or software of the client computer 28 is in turn sent to the host computer, and the latter then compiles an agent system adjusted thereto and sends it to the client computer 28. If, on the other hand, the test in step S23 reveals that the versions which can be loaded
10   from the host computer are not more up to date than the versions of the agent system, and components thereof, which are on the client computer, program execution is transferred to step S8, where the agent system is started. Hence, it is possible to update (during
15   operation) the entire agent system or, if required, merely parts thereof.

Steps S20 and S23 form an update function 31 which the agent system launcher 23a uses to update an existing
20   agent system.

The inventive method for loading an agent system is carried out by the agent system launcher 23a. The agent system launcher can be started manually at any time,
25   irrespective of whether or not an agent system is available or whether or not the agent system is in operation. The agent system launcher 23a can also be activated automatically when it is first loaded onto a computer or when a computer is started. If a full agent
30   system already exists, then the agent system launcher 23a can be controlled directly by the user or by the administration program 22, with the administration program 22 being able to call it repeatedly, thus ensuring that the agent system is updated regularly and
35   according to requirements.

To manage the agent systems on a plurality of mobile

client computers 28, a host computer 29 permanently connected to a network can be provided. If, by way of example, a small mobile computer, the client computer 28, having

5

a black and white screen is to start an agent system, then the host computer 29 sends a request to the client computer 28 and in so doing also transfers the current data relating to its resources (e.g. b/w screen, 8 MB

5   RAM, maximum number of windows which can be shown on the screen, Java Virtual Machine available, graphical display of characters, user management functionality in the agent system for one or more users, etc.). The host computer 29 uses this information to compile an agent

10 system which contains, by way of example, a Java class for printing specifically on b/w appliances. The equivalent class for printing on appliances with a color screen or a screen which can show only ASCII characters would have a different appearance. Depending

15 on a Java Virtual Machine (Persona Java, JDK1.x, 1.2, etc.) available on the client computer 28, the appropriate classes are compiled automatically. If only one user needs to be managed, as compared with a plurality, the appropriate class providing the user

20 management is also correspondingly smaller and has a smaller resource requirement on the client computer 28.

The inventive agent system can optionally be provided with a function which automatically tests the hardware

25 and/or software of the respective computer and starts the launcher 23a if required. This means that the agent system can be adjusted dynamically to changes in the hardware and/or in the software. This function can be provided by step S24 in figure 3, which follows step

30 S17. Step S24 tests whether the hardware and/or software has been changed since the agent system was last updated. If such a change is established, program execution passes directly to step S3, where an appropriate identifier is sent to the host computer 29.

35 If such a change is not established, the program passes to step S20 and is executed in line with the description above.

The programs on which the inventive agent systems are based can be stored and sold on electronically readable data stores.

Patent Claims

1.   An   agent   system   which   can   be   installed   on
computers connected in a computer network, having

5   -   an   agent   platform   (3)   which   comprises   service
programs required by an agent (25) in order to be able
to be executed on the respective computer (1),

-      an agent system controller (21) which has an agent
system launcher (23a) which can be loaded onto a client

10   computer   (28)   independently   of   the   rest   of   the   agent
system and is used for loading the agent system from a
host computer (29), characterized

in that the agent system launcher (23a) is designed
such that, before an agent system is loaded from a host

15   computer (29) onto the client computer (28) on which at
least the agent system launcher (23a) is installed, an
identifier identifying the hardware and/or software of
the client computer (28) is sent to the host computer
(29), and

20   an agent system update program (23b) installed on the
host computer (29) is designed such that it takes the
identifier identifying the hardware and/or the software
of the client computer (28) as a basis for compiling an
agent system adjusted to the hardware and/or software

25   of   the   client   computer   (28)   and   transmits   this   agent
system to the client computer (28).

2.   The agent system as claimed in claim 1,
characterized

30   in that the agent system launcher (23a) is designed
both for loading a full agent system (new installation
of an agent system) and for loading parts of an agent
system (update of an agent system).

35   3.   The agent system as claimed in claim 1 or 2,
characterized

in that the agent system launcher (23a) has a function for testing the hardware and/or software of the client computer (28) in order to produce the appropriate identifier.

5

4.   The agent system as claimed in one of claims 1 to 3, characterized
in that the agent system launcher (23a) has a function for comparing the version identifier of an agent system
10   installed on the client computer (28) with the version identifier of the agent system which can be loaded from the host computer (29).

5.   The agent system as claimed in one of claims 1 to 4,
15   characterized
in that the agent system is an agent system for mobile agents.

6.   A computer network having a plurality of computers
20   networked by means of data lines,
characterized by
an agent system having the features of one of claims 1 to 5 which is installed on a plurality of computers (1) in the computer network (9).

25

7.   The computer network as claimed in claim 6, characterized
in that at least one of the computers (1) is a mobile computer.

30

8.   The computer network as claimed in claim 7, characterized
in that the mobile computer is connected to the computer network (9) by means of a nonpermanent data
35   line (2).

9.   A method for loading an agent system from a host

computer (29) onto a client computer (28) in a computer
network (9), where an agent system update program (23b)
is installed on the host computer (29) and at least one

agent system launcher (23a) is installed on the client computer (28), the agent system launcher (23a) on the client computer (28) being able to load an agent system from the host computer (29) onto the client computer (28),

characterized

in that, before loading the agent system, the agent system launcher (23a) sends an identifier identifying the hardware and/or software of the client computer to the host computer (29), and

in that the agent system update program (23b) of the host computer (29) takes the identifier identifying the hardware and/or the software of the client computer (28) as a basis for compiling an agent system adjusted to the hardware and/or software of the client computer (28) and transmits this agent system to the client computer (28).

10. The method as claimed in claim 9,

characterized

in that the agent system launcher (23a) first checks whether an agent system is already installed on the client computer (28), and,

if an agent system is installed, tests (S15) whether an update is possible or required and whether the agent system is in operation (S16, S17), and

if an update is possible or required and the agent system is in operation, the operation of the agent system is terminated (S18) so that it can be updated.

11. The method as claimed in claim 9 or 10,

characterized by

a function (30) for loading an agent system from a host computer, where the call thereto involves the client computer (28) sending (S3) an identifier to the host computer (28) and then receiving (S7) the agent system sent by the host computer (29).

12.   The method as claimed in claim 11,

characterized by

a function (31) for updating the agent system, which function determines whether an update needs to be performed if an agent system is already installed on

5   the client computer (28).

13. The method as claimed in claim 12,

characterized

in that the function (31) for updating the agent system

10   first tests whether an update is possible in principle and, if an update is possible, it sends a version identifier to the host computer (29), which sends to the client computer (28) version information for the most recent agent system which can be loaded from the

15   host computer (29), the client computer then compares (S14) the version identifier with the version information and, if the comparison reveals that the version which can be loaded from the host computer (29) is more up to date than the version of the agent system

20   which is on the client computer (28), program execution is transferred to the function for loading the agent system.

14. The method as claimed in claim 13,

25   characterized

in that the comparison involves testing whether the individual parts of the agent system which can be retrieved from the host computer (29) are more up to date than those on the client computer (28), in which

30   case possibly only individual parts of the agent system are updated by the loading function (30).

15. The method as claimed in one of claims 8 to 14,

characterized

35   in that the identifier comprises a statement about the size of the memory of the client computer, of the type of display, the typical processing speed of the type of

network which can be used and/or of the type of system
software used.

GR 99 P 2597

Abstract

The invention relates to an agent system for mobile agents, to a computer network and to a method for loading an agent system from a host computer onto a client computer in a computer network.

The invention is distinguished in that the agent system which can be loaded onto the respective computer in the network is adjusted in line with the hardware and/or software of the computer. This makes it possible to incorporate even small mobile computers, in particular simple process computers, into a network in which agent systems are installed on the individual computers, without the power of the agent system needing to be significantly restricted in principle and without needing to dispense with the fundamental applicability of agent systems for small computers.

GR 99 P 2597

FIG 1

GR 99 P 2597

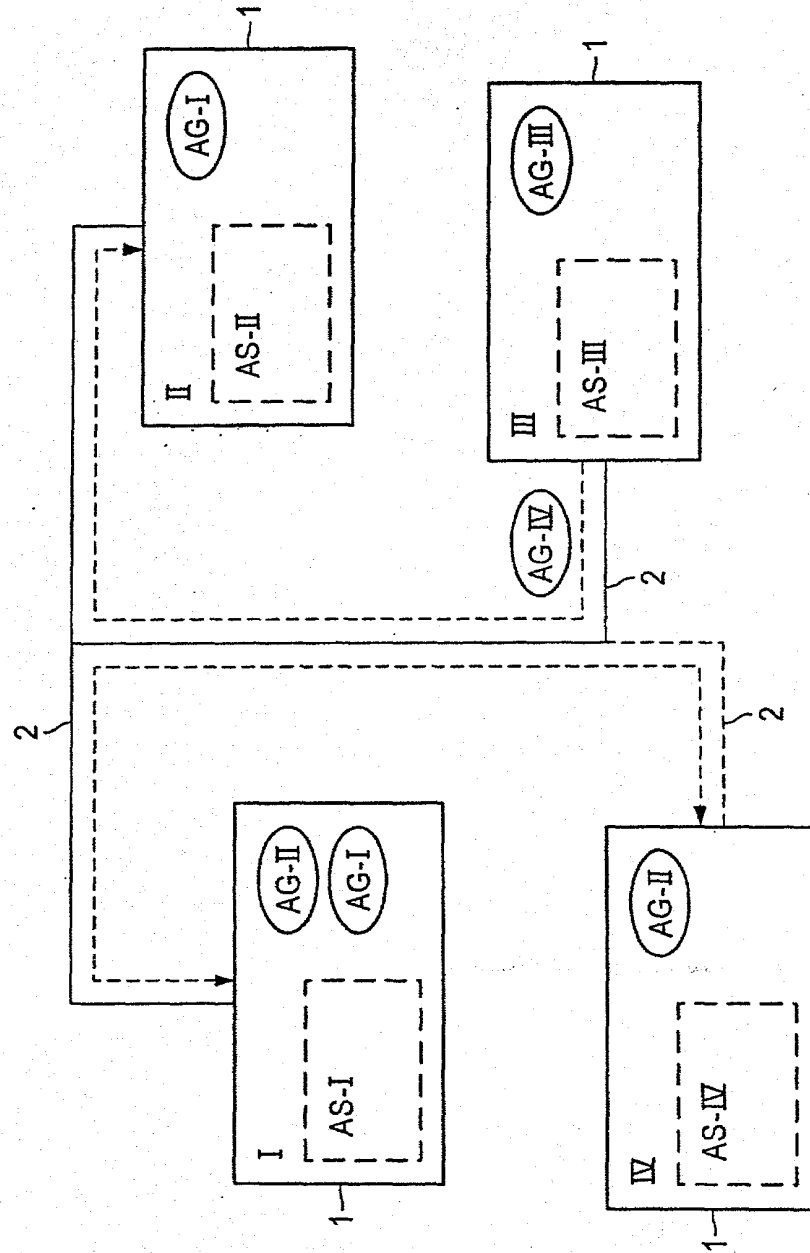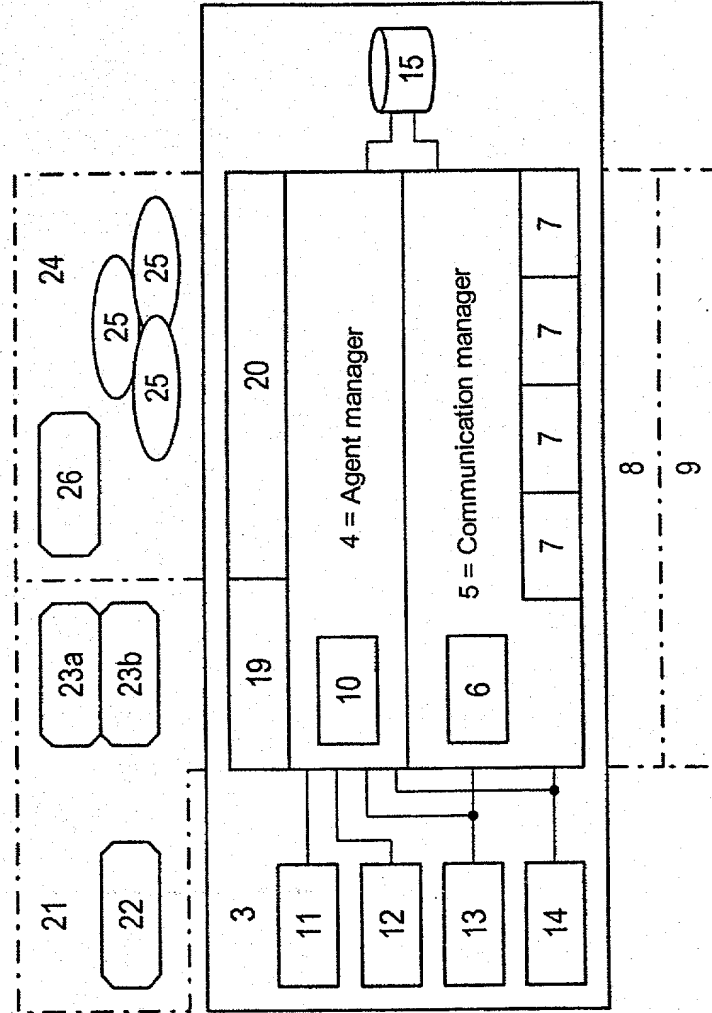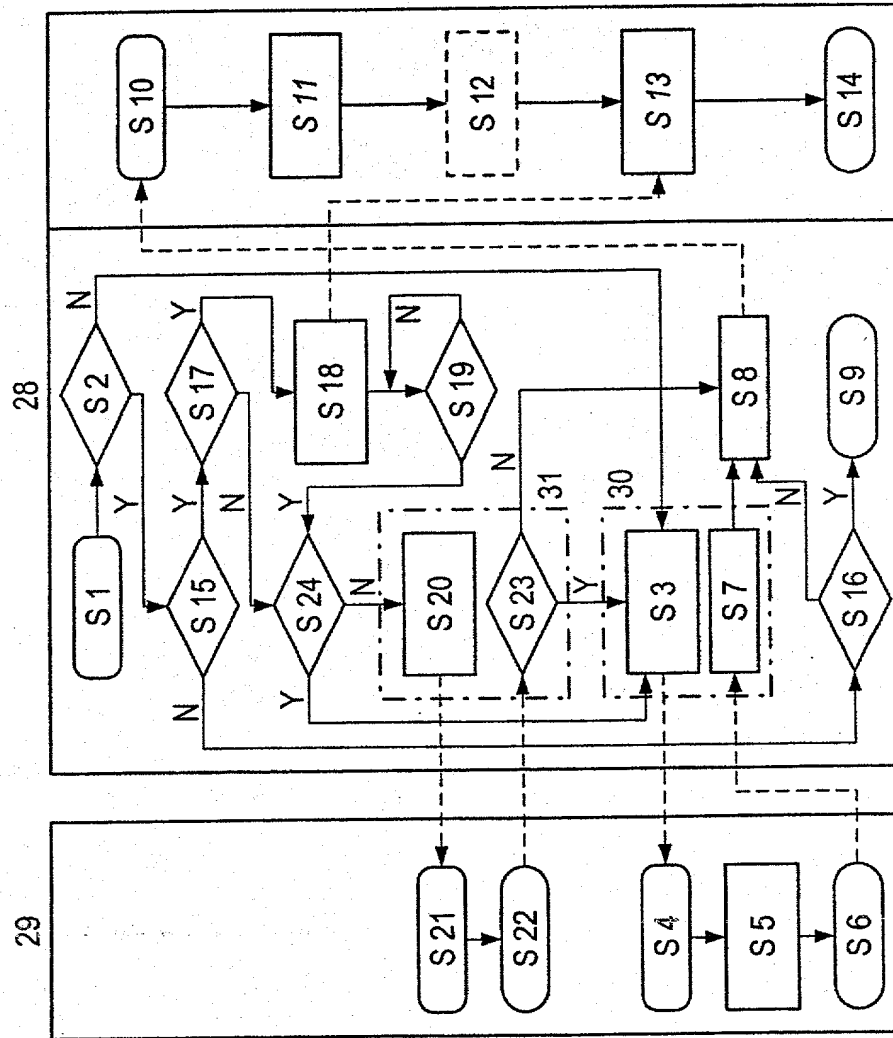FIG 2

GR 99 P 2597

3/3

FIG 3

Declaration and Power of Attorney For Patent Application
Erklärung Für Patentanmeldungen Mit Vollmacht
German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

As a below named inventor, I hereby declare that:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

My residence, post office address and citizenship are as stated below next to my name,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für den dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

<u>Agentensystem, Computernetzwerk und Verfahren zum Laden eines Agentensystems von einem Host-Computer auf einen Client-Computer eines Computernetzwerkes</u>

<u>Agent system for mobile agents, computer network and method for downloading an agent system from a host computer to a client computer of a computer network</u>

deren Beschreibung

the specification of which

(zutreffendes ankreuzen)
☐ hier beigefügt ist.
☒ am <u>11.08.2000</u> als
PCT internationale Anmeldung
PCT Anmeldungsnummer <u>PCT/DE00/02705</u>
eingereicht wurde und am _____
abgeändert wurde (falls tatsächlich abgeändert).

(check one)
☐ is attached hereto.
☒ was filed on <u>11.08.2000</u> as
PCT international application
PCT Application No. <u>PCT/DE00/02705</u>
and was amended on _____
(if applicable)

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die für die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Patent and Trademark Office-U.S. DEPARTMENT OF COMMERCE

IDNR: 2590 / V: 99-1.00 / B:Val

German Language Declaration

<u>19939058.4</u>	<u>DE</u>	<u>18.08.1999</u>	☒	☐
(Number)	(Country)	(Day Month Year Filed)	Yes	No
(Nummer)	(Land)	(Tag Monat Jahr eingereicht)	Ja	Nein

			☐	☐
(Number)	(Country)	(Day Month Year Filed)	Yes	No
(Nummer)	(Land)	(Tag Monat Jahr eingereicht)	Ja	Nein

			☐	☐
(Number)	(Country)	(Day Month Year Filed)	Yes	No
(Nummer)	(Land)	(Tag Monat Jahr eingereicht)	Ja	Nein

Ich beanspruche hiermit gemäss Absatz 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 120, den Vorzug aller unten aufgeführten Anmeldungen und falls der Gegenstand aus jedem Anspruch dieser Anmeldung nicht in einer früheren amerikanischen Patentanmeldung laut dem ersten Paragraphen des Absatzes 35 der Zivilprozeßordnung der Vereinigten Staaten, Paragraph 122 offenbart ist, erkenne ich gemäss Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) meine Pflicht zur Offenbarung von Informationen an, die zwischen dem Anmeldedatum der früheren Anmeldung und dem nationalen oder PCT internationalen Anmeldedatum dieser Anmeldung bekannt geworden sind.

I hereby claim the benefit under Title 35. United States Code. §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §122, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occured between the filing date of the prior application and the national or PCT international filing date of this application.

<u>PCT/DE00/02705</u>	<u>11.08.2000</u>	<u>anhängig</u>	<u>pending</u>
(Application Serial No.)	(Filing Date D, M, Y)	(Status)	(Status)
(Anmeldeseriennummer)	(Anmeldedatum T, M, J)	(patentiert, anhängig, aufgegeben)	(patented, pending, abandoned)

(Application Serial No.)	(Filing Date D,M,Y)	(Status)	(Status)
(Anmeldeseriennummer)	(Anmeldedatum T, M; J)	(patentiert, anhängig, aufgeben)	(patented, pending, abandoned)

Ich erkläre hiermit, dass alle von mir in der vorliegenden Erklärung gemachten Angaben nach meinem besten Wissen und Gewissen der vollen Wahrheit entsprechen, und dass ich diese eidesstattliche Erklärung in Kenntnis dessen abgebe, dass wissentlich und vorsätzlich falsche Angaben gemäss Paragraph 1001, Absatz 18 der Zivilprozessordnung der Vereinigten Staaten von Amerika mit Geldstrafe belegt und/oder Gefängnis bestraft werden koennen, und dass derartig wissentlich und vorsätzlich falsche Angaben die Gültigkeit der vorliegenden Patentanmeldung oder eines darauf erteilten Patentes gefährden können.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

German Language Declaration

VERTRETUNGSVOLLMACHT: Als benannter Erfinder beauftrage ich hiermit den nachstehend benannten Patentanwalt (oder die nachstehend benannten Patentanwälte) und/oder Patent-Agenten mit der Verfolgung der vorliegenden Patentanmeldung sowie mit der Abwicklung aller damit verbundenen Geschäfte vor dem Patent- und Warenzeichenamt: *(Name und Registrationsnummer anführen)*

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

And I hereby appoint

Customer No. 21171

Telefongespräche bitte richten an: *(Name und Telefonnummer)*	Direct Telephone Calls to: *(name and telephone number)*
	Ext. _____
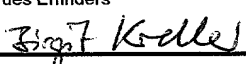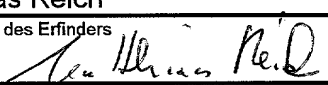
Postanschrift: Send Correspondence to:

Staas & Halsey LLP
700 Eleventh Street NW, Suite 500 20001 Washington, DC
Telephone: (001) 202 434 1500 and Facsimile (001) 202 434 1501
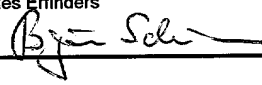or
Customer No. 21171

Voller Name des einzigen oder ursprünglichen Erfinders:	Full name of sole or first inventor:
Birgit Kreller 1 – ∞	Birgit Kreller
Unterschrift des Erfinders Datum	Inventor's signature Date
Birgt Krelle 30. 1.02	
Wohnsitz	Residence
Muenchen, DEUTSCHLAND	Muenchen, GERMANY DE
Staatsangehörigkeit	Citizenship
DE	DE
Postanschrift	Post Office Address
Gabelsbergerstr. 103	Gabelsbergerstr. 103
80333 Muenchen	80333 Muenchen
Voller Name des zweiten Miterfinders (falls zutreffend):	Full name of second joint inventor, if any:
Matthias Reich 2 – ∞	Matthias Reich
Unterschrift des Erfinders Datum	Second Inventor's signature Date
Matthias Reich 29. 1.02	
Wohnsitz	Residence
Muenchen, DEUTSCHLAND	Muenchen, GERMANY DE
Staatsangehörigkeit	Citizenship
DE	DE
Postanschrift	Post Office Address
Hofangerstr. 104	Hofangerstr. 104
81735 Muenchen	81735 Muenchen

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).

Page 3

Voller Name des dritten Miterfinders:	Full name of third joint inventor:
Dr. Bjoern Schiemann 3 - oo	Dr. Bjoern Schiemann
Unterschrift des Erfinders Datum	Inventor's signature Date
(signature) 29./.02	
Wohnsitz	Residence
Hoehenkirchen-Siegertsbr, DEUTSCHLAND	Hoehenkirchen-Siegertsbr, GERMANY DEX
Staatsangehörigkeit	Citizenship
DE	DE
Postanschrift	Post Office Address
Ottobrunner Str. 60	Ottobrunner Str. 60
85635 Hoehenkirchen-Siegertsbr	85635 Hoehenkirchen-Siegertsbr

Voller Name des vierten Miterfinders:	Full name of fourth joint inventor:
Unterschrift des Erfinders Datum	Inventor's signature Date
Wohnsitz	Residence
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

Voller Name des fünften Miterfinders:	Full name of fifth joint inventor:
Unterschrift des Erfinders Datum	Inventor's signature Date
Wohnsitz	Residence
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

Voller Name des sechsten Miterfinders:	Full name of sixth joint inventor:
Unterschrift des Erfinders Datum	Inventor's signature Date
Wohnsitz	Residence
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).   *(Supply similar information and signature for third and subsequent joint inventors).*

Form PTO-FB-240 (8-83)   Patent and Trademark Office-U.S. DEPARTMENT OF COMMERCE